N PAGE

Form Approved
OMB No. 0704-0188

2

Public rep
gathering
collection
Davis Higl

hour per response, including the time for reviewing instructions, searching existing data sources, ection of information. Send comments regarding this burden estimate or any other aspect of this ngton Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson ment and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGE

3. REPORT TYPE AND DATES COVERED
FINAL/01 SEP 91 TO 31 AUG 93

4. TITLE AND SUBTITLE
CAN WE BREAK INTRACTABILITY USING RANDOMIZATION
OR THE AVERAGE CASE SETTING? (U)

5. FUNDING NUMBERS
2304/A2
AFOSR-91-0347

6. AUTHOR(S)
Professor Joseph Traub

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Computer Science
Columbia University
450 Computer Science Building
New York, NY 10027

8. PERFORMING ORGANIZATION REPORT NUMBER
AEOSR-TR- 94 0245

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
110 DUNCAN AVE, SUITE B115
BOLLING AFB DC 20332-0001

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFOSR-91-0347

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE: DISTRIBUTION IS UNLIMITED

12b. DISTRIBUTION CODE
UL

13. ABSTRACT *(Maximum 200 words)*

The following papers cover results of the researchers and make up the final report: (1) A Surprising and Important New Result, by J F Traub, Feb 25, 1994, (2) Recent Progress in Information-Based Complexity, by J F Traub and H Wozinakowski, Invited Paper, Bulletin European Assoc for Theoretical Computer Science, Oct 1993, Number 51, pages 141-154 and (3) Breaking Intractability, by J F Traub and H Wozniakowski, published as cover story of Scientific American, Jan 1994.

14. SUBJECT TERMS

15. NUMBER OF PAGES
16

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR(SAME AS REPORT)

Can We Break Intractability Using Randomization or the Average Case Setting?

AFOSR-91-0347

Final Report

J. F. Traub

Computer Science Department
Columbia University

September 1, 1991 to September 30, 1993

Our results fall into the three major areas described below.

I. Breaking Intractability

Since I have kept AFOSR well informed of progress in this area I will not repeat myself here.

The following five papers and reports deal with progress in this area.

1. **A Surprising and Important New Result.** Report to AFOSR by J. F. Traub, February 25, 1993.
2. **Recent Progress in Information-Based Complexity**. J. F. Traub and H. Wozniakowski. Invited paper, Bulletin European Association for Theoretical Computer Science, October 1993, Number 51, pages 141-154.
3. **Breaking Intractability**. J. F. Traub and H. Wozniakowski. Published as cover story *Scientific American* January 1994.
4. **Development and Testing of Software for Multivariate Integration**. Report to AFOSR by S. Paskov and J. F. Traub, January 4, 1994.
5. **Tractability and Strong Tractability of Linear Multivariate Problems.** H. Wozniakowski. To be published in the March 1994 issue of the *Journal of Complexity*

We briefly describe the contents of the above papers and reports

- Item #1 is a report to AFOSR introducing the concept of strong tractability.
- Item #2 is an invited article which reviews recent progress in information-based complexity.
- Item #3 is an invited article for *Scientific American* which reports on recent progress in breaking intractability.
- Item #4 is a report to AFOSR on the status of development and testing of software for multivariate integration.
- Item #5 is the first publication regarding strong intractability. It will appear in the March 1994 issue of the *Journal of Complexity*.

II. Monte Carlo

The Monte Carlo Algorithm With A Pseudorandom Generator. J. F. Traub and H. Wozniakowski. Published in Mathematics of Computation, January 1992, Vol. 58, pages 323-339.

The current method of choice for computing multivariate integrals is Monte Carlo. Of course, on a computer there are no random numbers, only pseudo random numbers. There is a huge literature on statistical testing of pseudo random numbers. However these tests do not answer the question of most interest to the user. Are the good properties of the Monte Carlo algorithm using random numbers preserved if pseudo-random numbers are used? In this paper, which we believe to be the first on this topic, we prove that the answer is yes provided some care is taken. For example, in d dimensions it is necessary to use d random seeds.

III. Ill-Posed Problems

Linear Ill-Posed Problems Are Solvable On The Average For All Gaussian Measures. J. F. Traub and A. G. Werschulz. To appear, Math Intelligencer, 1994.

It has been proven that ill-posed problems are unsolvable in the worst -case deterministic setting. Yet ill-posed problems, which occur in many applications, must often be solved.

An answer may be provided in this paper. We show that ill-posed problems are solveable on the average for every Gaussian measure. This is the first paper on the average case analysis of ill-posed problems.

A SURPRISING AND IMPORTANT NEW RESULT

J. F. Traub

Computer Science Department
Columbia University

February 25, 1993

The number of function evaluations sufficient to solve important problems such as multivariate integration and multivariate approximation is completely independent of the number of variables!

CONTEXT FOR THE NEW RESULT

The following bullets put this new result into context.

- High-dimensional problems occur in numerous applications in science and engineering.

- Most of these problems cannot be solved analytically. They have to be numerically solved, approximately.

- Most multivariate problems are <u>intractable</u> in dimension. A typical result is that if accuracy ε is desired and there are d variables, then the computational complexity is $(1/\varepsilon)^d$.

- Thus, if a two-place answer is desired, the problem is 100 times harder for each additional variable. If eight-place accuracy is desired, the problem is 100,000,000 times harder for each additional variable.

- Although the physicists at Los Alamos did not know about computational complexity, they realized they could not solve certain problems. This led to the invention of Monte Carlo methods. For example, the computational complexity of multivariate integration in the randomized setting is proportional to $1/\varepsilon^2$ and therefore tractable.

- It was shown in 1989 that Monte Carlo methods cannot be used to break intractability of multivariate approximation.

- An alternative to the randomized setting is the average case setting in which we seek to break unsolvability and intractability by replacing a worst case guarantee that the error is less than the threshold ε with the weaker guarantee that the expected error is less than ε. Note that this is a deterministic setting; one has to solve the problem of optimal sample points.

- In 1991 it was shown that multivariate integration is tractable on the average. On a power scale, the average computational complexity of multivariate integration is proportional to $1/\varepsilon$. For small ε this is a major improvement over Monte Carlo, although for a different error criterion. Optimal sample points were obtained.
- Are other important multivariate problems tractable on the average? In 1992 it was shown that approximation is also tractable on the average. On a power scale the average computational complexity of multivariate approximation is proportional to $1/\varepsilon^2$. Optimal sample points were obtained.
- In the result stated above we ignored a multiplicative factor depending on the dimension d. For example, the average computational complexity of multivariate integration is $g(d)1/\varepsilon$, where $g(d)$ is a multiplacative factor which depends only on the number of variables. Good theoretical estimates of $g(d)$ are not known and obtaining them is believed to be very hard.

THE NEW RESULT

- An entirely new approach can be used. We get rid of the factor $g(d)$.
- Specifically, we say that a problem is strongly tractable if the number of function evaluations needed for the solution is completely independent of the number of variables. It depends only on a power of $1/\varepsilon$.
- This seems too much to ask for, but both multivariate integation and multivariate appoximation are strongly tractable on the average!
- This result is so new that it has not yet been written up.
- The result is given by a theorem and is non-constructive! That is, we know there must exist evaluation points in d dimensions which make integration and approximation strongly tractable, but these points are not yet known.

FUTURE RESEARCH

An exciting new result suggest new questions and directions, some of which we list here.

- What are the points of evaluation which make multivariate integration and multivariate approximation strongly tractable? This is a major challenge.
- We are currently implementing and testing software for multivariate integration using the known points which make this problem tractable on the average (but not strongly tractable).
- We then plan to implement and test this software for a network of workstations.
- We also plan to implement and test software for multivariate approximation.
- It has been shown that multivariate integration and multivariate approximation are strongly tractable. What other problems are strongly tractable?

RECENT PROGRESS
IN
INFORMATION–BASED COMPLEXITY

J. F. Traub[1] H. Woźniakowski[1,2]

TR-93-052

September, 1993

This is an invited article for the Structural Complexity Column, edited by Juris Hartmanis, which ~~will~~ appear ed in the Bulletin EATCS in October 1993. The scope of the article is indicated in the following list of Sections:

1. Overview of Information-Based Complexity
2. Breaking Intractability
3. Verification
4. Combinatorial Complexity
5. Similarities and Differences with Discrete Complexity
6. Brief History
7. Appendix
8. References

[1] Department of Computer Science, Columbia University, New York, New York 10027.
[2] Institute of Applied Mathematics, University of Warsaw, Banacha 2, 02-097 Warsaw, Poland.

The research reported here was supported in part by the National Science Foundation and the Air Force Office of Scientific Research. .

Typeset by $\mathcal{A}\mathcal{M}\mathcal{S}$-TEX

1. Overview of Information-Based Complexity

The goal of this article is to report some of the recent progress in information-based complexity, which for brevity will be denoted as IBC. We have selected topics which might be of particular interest to the EATCS audience. We take an informal approach in this article, focusing mainly on ideas. For precise formulations and results, as well as proof techniques, see the books TW[1][80], TWW [83], Novak [88], TWW [88], Werschulz [91], and recent surveys , PT [87], PW [87], TW [91a, 91b], Heinrich [92], and Novak [93].

We begin by presenting a greatly simplified picture of computational complexity to indicate where IBC fits in. For our present purpose, computational complexity may be divided into two branches, discrete and continuous. Continuous computational complexity may again be split into two branches. The first, which we'll call *continuous combinatorial complexity*, deals with problems for which the information is *complete*. Problems where the information may be complete are those which are specified by a finite number of parameters. Examples include linear algebraic systems, matrix multiplication, and systems of polynomial equations. Blum, Shub and Smale [89] obtained the first NP-completeness results over the reals for a problem with complete information.

The other branch of continuous computational complexity is IBC. Typically, IBC studies infinite-dimen-sional problems. These are problems where either the input or the output are elements of infinite-dimensional spaces. Since digital computers can handle only finite sets of numbers, infinite-dimensional objects such as functions on the reals must be replaced by finite sets of numbers. Thus, complete information is not available about such objects. Only *partial* information is available when solving an infinite-dimensional problem on a digital computer. Typically, information is *contaminated* with errors such as round-off error, measurement error, and human error. Thus, the available information is partial and/or contaminated.

We want to emphasize this point for it is central to IBC. *Since only partial and/or contaminated information is available, we can solve the original problem only approximately. A goal of IBC is to obtain the computational complexity of computing such an approximation.*

In Figure 1 we schematized the structure of computational complexity described above.

[1] When one of us is a co-author, the citation will be made using only initials

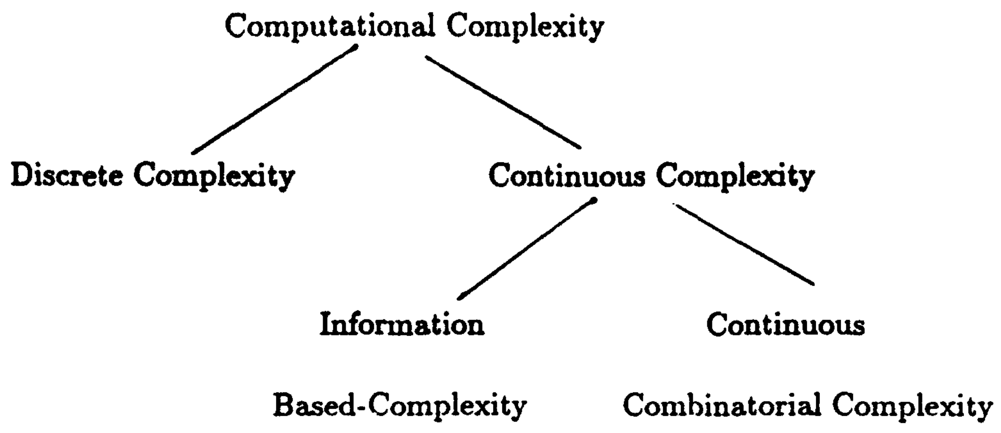


Figure 1

The motivation for studying IBC is two-fold:

(1) Continuous models, typically infinite-dimensional, are very common in science, engineering, economics, and even in finance. Examples of the mathematical problems which arise from these models are partial or ordinary differential equations, multivariate integration, and optimization.

(2) The subject matter covered by IBC is rich from a complexity point of view with many results and numerous open questions, as we hope to illustrate in this article.

Although IBC typically studies infinite-dimensional problems there are important exceptions. These include probabilistic complexity of processor synchronization with stochastic delays, Wasilkowski [88a], and complexity of solving large linear systems, TW [84], Nemirovsky [91, 92].

IBC is formulated as an abstract theory; see the Appendix. The applications often involve multivariate functions over the reals. For example, in multivariate integration, the integrand is a multivariate function. In optimization, one seeks an extremum of a multivariate function subject to multivariate constraints. In an initial-value problem, such as the wave equation, the initial condition is again specified by a multivariate function.

The observation that a function over the reals cannot be entered into a digital computer lies at the heart of IBC. (In the general case, an element of an abstract space cannot be entered.) We call a multivariate function *a mathematical input*, denoted by $\mathrm{I}_{\mathrm{math}}$. Let S be a linear or nonlinear operator which specifies the problem we want to solve, $S : F \to G$

for some sets F and G. The operator S carries $\mathrm{I}_{\mathrm{math}}$ from F into a mathematical output $\mathrm{O}_{\mathrm{math}}$ in G; see Figure 2(a)

$$\begin{tikzcd}
\mathrm{I}_{\mathrm{math}} \arrow[rr, "S"] & & \mathrm{O}_{\mathrm{math}}
\end{tikzcd}$$

Figure 2(a)

Of course, this is too general to characterize an IBC problem. For example, $\mathrm{I}_{\mathrm{math}}$ could be the locations of a set of cities and $\mathrm{O}_{\mathrm{math}}$ could be an optimal tour; which is a typical discrete problem. This is an IBC problem when $\mathrm{I}_{\mathrm{math}}$ cannot be entered into a digital computer, and it must be replaced by a computer input denoted by $\mathrm{I}_{\mathrm{comp}}$.

The computer input, $\mathrm{I}_{\mathrm{comp}}$, consists of a finite set of numbers. For example, if $\mathrm{I}_{\mathrm{math}}$ is a function then $\mathrm{I}_{\mathrm{comp}}$ might consist of its values at certain points. $\mathrm{I}_{\mathrm{comp}}$ is obtained from $\mathrm{I}_{\mathrm{math}}$ by *information operations*. Different disciplines have different names for these information operations. Computer scientists called them oracle calls, mathematicians call them functionals, and engineers call them black-box calls. The replacement of $\mathrm{I}_{\mathrm{math}}$ by $\mathrm{I}_{\mathrm{comp}}$ may be viewed as a discretization.

Denote the set of information operations by $N(\mathrm{I}_{\mathrm{math}})$; we call N the *information* operator. Since many (typically, an infinite number of) mathematical inputs map into the same computer input, the mapping N is many-to-one. That is, discretization is irreversible. The situation is diagrammed in Figure 2(b).

$$\begin{tikzcd}
\mathrm{I}_{\mathrm{math}} \arrow[rr, "S"] \arrow[d, "\mathrm{N}"'] & & \mathrm{O}_{\mathrm{math}} \\
\mathrm{I}_{\mathrm{comp}} & &
\end{tikzcd}$$

Figure 2(b)

Although there has been mention of neither computer output nor algorithm, we can already draw certain conclusions. Since N is a many-to-one map, the computer does not know the mathematical input. Therefore, it is impossible to solve this problem exactly; the best we can hope for is an approximation.

We assign the same cost to each information operation. Given an error threshold ε, we can define the information complexity, $\mathrm{COMP}^{\mathrm{info}}(\varepsilon)$, as the minimal cost of the information operations needed to obtain an ε-approximation. (In computational learning theory this is called sample complexity.) Information complexity can be defined in different settings such as the worst case, average case or probabilistic setting.

Using the concept of *radius of information*, $r(N)$, see TW [80, pp. 9-15], TWW [88, pp. 43-45, 197-200, 327-328], we can often obtain sharp lower and upper bounds on the information needed to get an ε-approximation. The information N is powerful enough to obtain an ε-approximation iff

$$r(N) \leq \varepsilon.$$

Since the information complexity is a lower bound on the computational complexity, defined below, this has led to *proven* (not conjectured) intractability and unsolvability results which we'll describe in Section 2.

Because of the basic role played by information-level results we decided to name this area information-based complexity. This level typically does not exist for discrete problems. However, combinatorial issues will play an increasingly important role in IBC: see Section 4.

Let the computer output be denoted by $\mathrm{O_{comp}}$ and the operator that maps $\mathrm{I_{comp}}$ into $\mathrm{O_{comp}}$ by ϕ. We call ϕ a combinatory algorithm (algorithm for brevity). Since ϕ maps the computer input into the computer output it plays the same role as algorithm does elsewhere in computer science. Figure 2(c) completes the picture.

$$\begin{tikzcd}
\mathrm{I_{math}} \arrow[rr, "S"] \arrow[d, dash, "N"'] & & \mathrm{O_{math}} \\
\mathrm{I_{comp}} \arrow[rr, "\phi"'] & & \mathrm{O_{comp}}
\end{tikzcd}$$

Figure 2(c)

Observe that $\mathrm{O_{comp}} \neq \mathrm{O_{math}}$ because N is many-to-one. In other words, S does not commute with ϕ composed with N.

We now discuss the model of computation used in IBC. For simplicity, we restrict ourselves to the case that $G = \mathbf{R}$. We assume that the real number model is chosen as our model of computation. (See Section 5 for a discussion of why the real number model is often used in IBC and also of research on finite models.) That is, we assume that arithmetic operations and comparisons on real numbers are carried out exactly and at unit cost.

We define the combinatorial complexity, $\mathrm{COMP^{comb}}(\varepsilon)$, as the minimal cost of the combinatory operations needed to compute an ε-approximation if all information operations were free.

Finally, we define the computational complexity, COMP(ε), as the minimal cost of computing the computer output with error at most ε under the assumption that information and combinatory operations are charged.

As before, combinatorial and computational complexity may be defined in the worst case, average case and probabilistic settings. Note that,

$$\mathrm{COMP}(\varepsilon) \geq \max\{\mathrm{COMP}^{\mathrm{info}}(\varepsilon), \mathrm{COMP}^{\mathrm{comb}}(\varepsilon)\}.$$

We conclude this overview by characterizing IBC and stating its major goals. IBC studies problems which have the properties listed below.

(1) $\mathrm{I}_{\mathrm{comp}} \neq \mathrm{I}_{\mathrm{math}}$.

(2) There is a charge for obtaining $\mathrm{I}_{\mathrm{comp}}$.

We discuss the first of these. These are two major reasons why $\mathrm{I}_{\mathrm{comp}} \neq \mathrm{I}_{\mathrm{math}}$. The first is that the mathematical input cannot be represented by a finite set of numbers. We say the information about $\mathrm{I}_{\mathrm{math}}$ is *partial*. An important example in applications is when $\mathrm{I}_{\mathrm{math}}$ is a multivariate function. A second reason is that the information about $\mathrm{I}_{\mathrm{math}}$ is *contaminated*. Information may be contaminated because of round-off or measurement errors.

We list some of the major goals of IBC.

(1) Obtain good lower and upper bounds on the computational complexity, information complexity, and combinatorial complexity.

(2) Find information N and an algorithm ϕ for which the computational complexity is attained or nearly attained. Such N and φ are called *optimal*, or *nearly optimal*.

We summarize the reminder of this article. We will present a selection of recent results from a number of IBC areas. We then conclude this article with a discussion of similarities and differences with discrete complexity and a brief history. An abstract formulation of IBC may be found in the Appendix.

2. Breaking Intractability

It has been established that in the worst case deterministic setting many problems studied in IBC are unsolvable or intractable. More precisely, let the mathematical input f be a multivariate function of d variables. Let the smoothness of the set of inputs be denoted by r. For example, we might require that all partial derivatives of f up to order r exist and are uniformly bounded by 1. Assume we want to guarantee an error at most ε. Then, for many continuous problems the worst case computational complexity, COMP(ε), is given by

$$\mathrm{COMP}(\varepsilon) = \Theta\left(\left(\frac{1}{\varepsilon}\right)^{d/r}\right). \tag{1}$$

For example, multivariate integration, function approximation, partial differential equations, integral equations, and nonlinear optimization all have this computational complexity, see Bakhvalov [59], Heinrich [93], Nemirovsky and Yudin [83], Novak [88], Pereverzev [89], TWW [88], and Werschulz [91].

Furthermore, many problems in science, engineering, economics and even finance use mathematical models with large d. For example, computational chemistry, computational design of pharmaceuticals, and computational metallurgy involve computation with large number of particles. Since the specification of each particle requires three variables for static problems and six variables for dynamic problems, this leads to problems with very large d. For path integrals, important in the foundation of physics, $d = +\infty$; they invite approximation by multivariate integration with huge d. Problems with large d are also important in mathematical disciplines such as statistics and geometry.

Observe that we can conclude that if the smoothness r is fixed and positive then the computational complexity is an exponential function in d. Thus, problems whose complexity is governed by (1) are intractable in d. If $r = 0$, that is, if the class of inputs is only continuous, then $\mathrm{COMP}(\varepsilon) = +\infty$ for small ε; that is, the problem is *unsolvable*.

The only way to break unsolvability or intractability is to weaken the assurance of an ε-approximation by shifting to another setting. Three settings have been used for trying to break intractability: randomized, average case, and probabilistic settings. Here we confine ourselves to recent advances on breaking intractability in the average case setting. See TW [91a] for a survey of how to break intractability in the randomized setting.

We describe recent advances in breaking intractability for multivariate integration and multivariate function approximation. Multivariate integration is especially common since computing the expectation of any stochastic process leads to this problem.

In the average case setting the average computational complexity, $\mathrm{COMP}^{\mathrm{avg}}(\varepsilon)$, is defined as the minimal expected cost such that the average error is less than ε. One has to put a measure on the space of inputs. Although for discrete problems one can assume that all inputs are equiprobable, no such assumption can be made for typical sets of functions. The most commonly used measures on function spaces are Gaussian measures, and, in particular, Wiener measures which are a special kind of Gaussian measure.

It was known that multivariate integration is tractable on the average but the proof is non-constructive. That is, the optimal points at which the integrand should be evaluated and the average computational complexity were unknown.

Then W [91] established a relation between discrepancy and the average complexity of multivariate integration. Discrepancy has been extensively studied in number theory and sharp bounds on discrepancy in d dimensions were established by Roth [54,80]. The use of the results from discrepancy theory solved the multivariate integration problem.

We describe the results more precisely. Let $r = 0$. Recall that in the worst case deterministic setting the problem is unsolvable. Assume the measure on the integrands is the Wiener sheet measure. Then

$$\mathrm{COMP}^{\mathrm{avg}}(\varepsilon) = \Theta\left(\frac{1}{\varepsilon}\left(\log\frac{1}{\varepsilon}\right)^{(d-1)/2}\right).$$

Thus a problem which is worst-case unsolvable becomes tractable on the average.[2] Either Hammersley points or hyperbolic-cross points are nearly optimal as the evaluation points in d dimensions. These results were generalized to the case of smooth inputs by Paskov [93].

We turn to the average complexity of function approximation. This is particularly important since unlike for multivariate integration, it is known that randomization does not help for function approximation, see Wasilkowski [88b], Novak [92]. Again, let $r = 0$ and assume a Wiener sheet measure. Then

$$\mathrm{COMP}^{\mathrm{avg}}(\varepsilon) = \Theta\left(\frac{1}{\varepsilon^2}\left(\log\frac{1}{\varepsilon}\right)^{2(d-1)}\right)$$

and again hyperbolic cross-points can be used; see W [92b].

Roth's discrepancy results and the average computational results quoted above are big theta results in ε. That is, the dependence on ε is known, but there is a multiplicative factor, $g(d)$, which is not known. If we're serious about solving problems with large d we must be able to bound $g(d)$. It is believed that obtaining good theoretical estimates of $g(d)$ is very hard.

The problem may be solved by getting rid of the factor $g(d)$ in the following way, W [93]. A problem is said to be ***strongly tractable*** if the number of information operations, $m(\varepsilon, d)$, needed to compute an ε-approximation is independent of d and depends polynomially on $1/\varepsilon$, that is,[3]

$$m(\varepsilon, d) \le K\left(\frac{1}{\varepsilon}\right)^p, \qquad \forall d,\ \forall \varepsilon \le 1,$$

[2] By tractable (in $1/\varepsilon$) we mean that the complexity is bounded by $K(d)\,(1/\varepsilon)^p$ for all d and $\varepsilon \le 1$ for a number p which is independent of d and ε.

[3] More precisely, it is required that the computational complexity can be bounded by $K\,c(d)\,(1/\varepsilon)^p$ for certain numbers K and p, independent of d and ε, where $c(d)$ is the cost of one information evaluation of a function of d variables.

for certain numbers K and p.

That might seem to much to expect but multivariate integration and multivariate approximation are both strongly tractable on the average[4] and it is sufficient to take the information operations as function evaluations, W [93]. Usually in computational complexity, an upper bound is given by an algorithm and a lower bound by a theorem. But in this case, the upper bound has been determined by a theorem and is *non-constructive*. That is, we know that there must exist sample points at which we should evaluate the function and a combinatory algorithm which make multivariate integration and approximation strongly tractable. The construction of such sample points and algorithm is being studied; WW [94].

Due to the relation between discrepancy and average case multivariate integration, strong tractability for multivariate integration implies that the discrepancy of n points in d dimensions can be bounded, independently of d, by Kn^{-p} with the same K and p for both problems. This estimate is of interest in its own right since discrepancy is of considerable interest in number theory, see Beck and Chen [87], and Niederreiter [92]. Furthermore there are numerous applications of discrepancy; for example, for applications in computer graphics, see Dobkin and Mitchell [93].

3. Verification

Most of IBC has been devoted to the computational complexity of computing an ε-approximation. Recently, the computational complexity of *verification* has been studied, that is checking whether an answer is correct, see W [92a]. In addition to being given a problem, we are also given an "answer" g and asked whether it is true that g is within ε of the mathematical output; see the Appendix for a precise definition.

The reader's reaction may be that, of course, verification is no harder than computation. Indeed, if the mathematical output can be computed exactly at finite cost, as is the case for discrete problems, then with one extra comparison one can solve the verification problem.

However, for typical IBC problems the mathematical output *cannot* be computed with finite cost, and the relation between verification and computation is not obvious. As we shall see, in the worst case setting verification may be unsolvable while the corresponding computational problem is easy.

We illustrate this with a simple example. The computational problem is to compute an ε-approximation to $\int_0^1 f(x)\,dx$ where the mathematical input f is an arbitrary function

[4] We stress that this holds for the Wiener sheet measure. For an isotropic Wiener measure, function approximation is still intractable even on the average, see Wasilkowski [93].

over $[0,1]$ satisfying a Lipschitz condition with constant at mcst one. The computational input is given by values of f at some points. The computational complexity in the worst case setting is known to be of order $1/\varepsilon$; thus the computational problem is "easy".

Suppose now that we're given the purported answer g and asked to check whether this is within ε of the integral of f. We show that the verification problem is unsolvable.

Suppose that we compute f at a finite number of points x_i and that for every such point $f(x_i) = g + \varepsilon$. If we answer NO the adversary will choose $f(x) \equiv g + \varepsilon$. This function is certainly Lipschitz (with constant zero), and compatible with the computed function values. Since $\int_0^1 f(x)\,dx = g + \varepsilon$ is within ε of the answer g, we made a mistake by answering NO.

If we answer YES the adversary will choose a hat function f going through the points $(x_i, g+\varepsilon)$ and with Lipschitz constant one. Clearly, $\int_0^1 f(x)\,dx > g+\varepsilon$ which is *not* within ε of the answer g. We made a mistake by answering YES. Hence, as long as we have finitely many function values, there is no way to solve the verification problem in the worst case setting.

It can be shown that verification for IBC problems is often unsolvable in the worst case setting. Verification is therefore studied in the probabilistic setting. Here we want to verify that g is an ε-approximation with confidence δ; see the Appendix. In this setting the probabilistic complexity of verification depends on how ε and δ are related. Any relation between the probabilistic complexities of verification and computation is possible. In particular, verification can be exponentially (in δ) harder than computation.

NW [92] studied *relaxed* verification in the worst case setting. That is the answers can be YES, NO, or DON'T CARE. The size of the DON'T CARE region is specified by a parameter α; see the Appendix. For a positive α, the worst case complexity of relaxed verification is finite. It is related to the worst case complexity of the computational problem with ε replaced by roughly $\varepsilon\,\alpha^q$ with $q \in [0,1]$ depending on the problem. Hence, if α is not too small, the complexity of relaxed verification is roughly comparable to the complexity of the computational problem. If, however, α is small then the complexity of relaxed verification is usually much larger than the complexity of the computational problem.

4. Combinatorial Complexity

To date, IBC problems have usually been proven unsolvable or intractable by showing that their *information complexity* was infinite or exponential. Recent results establish unsolvability or intractability by showing that the *combinatorial complexity* is infinite or, if P$\neq$NP, not polynomial. We report these results and also pose an open question.

Papadimitriou and Tsitsiklis [86] is a pioneering paper which proves that a nonlinear problem in decentralized control theory is intractable if P$\neq$NP. More precisely, the information complexity is a polynomial in $1/\varepsilon$ but the combinatorial complexity in a Turing machine model of computation is not polynomial in $1/\varepsilon$, if P$\neq$NP.

WW [93] show that there exists a linear problem whose information complexity is a polynomial in $1/\varepsilon$ but whose combinatorial complexity is infinite[5], making the problem unsolvable. An "artificial" problem is constructed to show that even a linear problem can be very hard combinatorially. Chu [94] shows that the combinatorial complexity can be any increasing function of the information complexity.

We pose an open question. So far, tight bounds on the computational complexity of IBC problems are achieved when the minimal amount of information is used. Is there a problem for which more information operations should be used to achieve the computational complexity? That is, does there exist a problem for which the minimal amount of information is very hard to combine but if more information operations are computed then it is easier to combine them and the total cost of computing an ε-approximation is minimized in the latter case.

We believe that in the future, progress in IBC will increasingly require results in both information complexity and combinatorial complexity.

5. Similarities and Differences with Discrete Complexity

We begin with similarities. As in the rest of computational complexity, IBC studies lower and upper bounds on the computational difficulty of solving mathematically posed problems. Optimal and near-optimal algorithms are sought. To attempt to break the intractability results and conjectures of the worst case deterministic setting, both IBC and discrete complexity turned to other settings such as the randomized and average case settings.

There are also significant contrasts, three of which we will discuss in the remainder of the section. IBC has the following characteristics:

Problems cannot be exactly solved

Intractability has been proven for many problems

Real number model usually used

We discuss each of these.

[5] This result holds if we allow arithmetic operations, comparisons of real numbers, and precomputation. It is open if there exists a linear problem with finite information complexity and infinite combinatorial complexity in the extended real number model in which logarithms, exponentials and ceilings are allowed.

Problems Cannot Be Exactly Solved

As discussed in Section 1, it is impossible to solve IBC problems exactly because $I_{comp} \neq I_{math}$. It is possible, in principle, to solve discrete problems exactly although one may choose to solve them approximately to reduce the cost.

Intractability has been proven for many problems

Using information-level arguments, unsolvability and intractability has been established for many IBC problems. With only a few exceptions, there are no non-trivial lower bounds on the combinatorial complexity of IBC problems. Since only combinatorial arguments are available, intractability of many discrete problems has been conjectured. (Of course, lower bounds, as well as unsolvability results, have been established for some combinatorial problems.)

Real number model usually used

To date, the real number model of computation has usually been used in continuous computational complexity. After discussing the motivation, we turn to finite models for continuous computational complexity.

Scientific problems are usually solved using fixed precision floating point arithmetic. The cost of floating point operations and comparisons is independent of the size of the operands. Furthermore, all arithmetic operations and comparisons cost about the same to execute. Our goal is to choose a model of computation that corresponds to performance of a digital computer executing floating point arithmetic. The abstraction we choose is the *real number model*, which assumes that arithmetic and comparisons on real numbers can be executed exactly and at unit cost. (The choice of unit cost is just scaling.) Rounding errors occur when a digital computer executes operations in fixed precision floating point arithmetic. In our abstraction we assume arithmetic is performed without error. This separation of complexity theory from error analysis is done for technical reasons; computational complexity theory is hard enough without including round-off error. When an interesting new algorithm is discovered from computational complexity considerations, a stability analysis in fixed precision floating point arithmetic must be performed.

We stress that the real number model is *not* polynomially equivalent to the Turing machine model. For example, TW [82] shows that the cost of Kachian's algorithm is not polynomial in the real number model and conjecture that linear programming is not polynomial in this model. This conjecture is still open.

Several finite models of computations have also been analyzed. One of them is a model based on recursive analysis, see Ko [91].

In the bit model it is assumed that one can get a rational binary approximation of a real number or of a function value to within any accuracy with the cost depending on the number of bits. This model has been studied for problems with *complete* information, for instance, for finding roots of polynomials, see Schönhage [86]. A mixed model, in which the bit model is used for information operations, and the real number model for combinatory operations, is utilized by Kacewicz and Plaskota [90] to analyze certain IBC problems.

It is, of course, desirable to fully explore finite models for IBC problems and we believe this to be an important direction for future research.

6. A Brief History

We present a very brief history of IBC. Research in the spirit of IBC was initiated in the Soviet Union by Kolmogorov in the late 40's. Nikolskij [50], a student of Kolmogorov, studied optimal quadrature. This line of research was greatly advanced by Bakhvalov, see e.g., Bakhvalov [59, 71]. In the United States research in the spirit of IBC was initiated by Sard [49] and Kiefer [53]. Kiefer reported the results of his 1948 MIT Master's Thesis that Fibonacci sampling is optimal when approximating the maximum of a unimodal function. Sard studied optimal quadrature. Golomb and Weinberger [59] studied optimal approximation of linear functionals. Schoenberg [64] realized the close connection between splines and algorithms optimal in the sense of Sard.

T[61,64] initiated the study of iterative computational complexity, emphasizing the central role of information. Maximal order results, needed to obtain lower bounds on computational complexity, were obtained for scalar nonlinear equations. W [75] introduced the concept of order of information in an abstract space which provides a general tool for establishing maximal order of an algorithm.

Micchelli and Rivlin [77] studied optimal recovery and considered optimal error algorithms for the approximation of linear operators. Linear noisy information was permitted.

A general formulation of IBC, primarily in the worst case deterministic setting, is presented in TW [80], where a somehow more detailed history and an annotated bibliography of over 300 papers and books up to 1979 can be also found. At the time IBC was called analytic complexity to differentiate it from algebraic complexity. TWW [88] extend the study of IBC to numerous settings including average case, randomized, probabilistic, and asymptotic settings, as well as mixed settings.

Appendix

We present an abstract formulation of IBC. Let

$$S : F \to G$$

where F is a subset of a linear space and G is a normed linear space.

For $f \in F$, we wish to compute an approximation to $S(f)$. To do this we must know something about f. A basic assumption is that we have only partial information about f. We gather this partial information about f by information operations $L(f)$. Here we will assume that L is a linear functional. Let Λ denote the class of information operations we will permit. The choice of Λ will depend on the problem we wish to solve. If we wish to approximate a definite integral we must exclude definite integration as a permissible information operation, and for this problem Λ is usually defined as the class of function evaluations. For other problems, such as the solution of nonlinear equations, we may permit any linear functional. Let

$$N(f) = [L_1(f), \ldots, L_n(f)],$$

for $L_i \in \Lambda$. Here L_i, as well as n, can be adaptively chosen depending on the already computed information operations.

$N(f)$ is called the *information* on f and N the *information operator*. The motivation for introducing the information operator N is to replace the element f, which is often from an infinite-dimensional space, by n numbers. An idealized algorithm[6] ϕ is an operator $\phi : N(F) \to G$. The approximation $U(f)$ is then computed by

$$U(f) = \phi(N(f)).$$

(The assumption that the approximation is the composition of ϕ with N is made without loss of generality.) We seek $U(f)$ such that

$$\|S(f) - U(f)\| \le \varepsilon.$$

We say $U(f)$ is an *ε-approximation.*

We illustrate the abstract model by an integration example with

$$S(f) = \int_0^1 f(t)dt,$$

[6] By using such a general definition of algorithm, we strengthen the lower bound conclusions. For upper bounds, we restrict the algorithms to those constructed from permissible combinatory operations.

$$F = \{f : \ f \in C^r(0,1) \text{ and } \|f\|_{\max} \le 1\},$$

and G as the set of real numbers. The functionals are chosen as $L_i(f) = f(t_i)$. An example of an algorithm is

$$U(f) = \phi(N(f)) = \frac{1}{n}\sum_{i=1}^{n} f(t_i).$$

To define computational complexity we must first introduce our model of computation, which is defined by two postulates:

(i) Let Ω denote the set of permissible combinatory operations including the addition of two elements in G, multiplication by a scalar in G, arithmetic operations on real numbers, and comparison of real numbers. We assume that each combinatory operation is performed exactly with unit cost.

(ii) We assume that we are charged for each information operation. That is, for every $L \in \Lambda$ and $f \in F$, the computation of $L(f)$ costs c, where $c > 0$. Typically, $c \gg 1$.

We assume the ***real number model***, that is, we can perform operations on real numbers exactly and at unit cost. See Section 5 for a discussion and motivations underlying the model of computation and the real number model.

We briefly describe how the computation is carried out and how its cost is calculated. Let cost(N, f) denote the cost of computing the information $N(f)$. Knowing the information $N(f)$, the approximation $U(f) = \phi(N(f))$ is computed by combining the information to produce an element of G which approximates $S(f)$.

Let cost$(\phi, N(f))$ denote the cost of computing $U(f) = \phi(N(f))$, given $N(f)$. Then the total cost of computing $U(f)$, cost(U, f), is

$$\text{cost}(U,f) = \text{cost}(N,f) + \text{cost}(\phi, N(f)).$$

We are ready to define the computational complexity, comp(ε), as

$$\text{comp}(\varepsilon) = \inf\{\text{cost}(U) : U \text{ such that } e(U) \le \varepsilon\},$$

with the convention that $\inf \emptyset = \infty$. The definition of cost(U) and $e(U)$ varies according to the setting. Settings studied in IBC include worst case, average case, probabilistic, randomized and asymptotic. Mixed settings are also studied. We confine ourselves here to the definition of just the worst case and average case settings.

Worst Case Setting: The *worst case error* and *worst case cost* of U are defined by

$$e(U) = \sup_{f \in F} \|S(f) - U(f)\|,$$
$$\text{cost}(U) = \sup_{f \in F} \text{cost}(U,f).$$

Average Case Setting: Let μ be a probability measure defined on F. The ***average case error*** and ***average case cost*** of U are defined by

$$e(U) = \sqrt{\int_F \|S(f) - U(f)\|^2 \, \mu(df)},$$
$$\mathrm{cost}(U) = \int_F \mathrm{cost}\,(U, f)\mu(df).$$

The concept of complexity permits us to introduce the fundamental concepts of optimal information and optimal algorithm. Information N and an algorithm ϕ that uses N are called ***optimal information*** and ***optimal algorithm***, respectively, iff $U = \phi \cdot N$ satisfies $\mathrm{cost}(U) = \mathrm{comp}(\varepsilon)$ and $e(U) \le \varepsilon$.

We define the verification problem. For given $g \in G$ we want to check whether $\|S(f) - g\| \le \varepsilon$. That is, we define $\mathrm{VER}(f,g)$ =YES if $\|S(f) - g\| \le \varepsilon$, and $\mathrm{VER}(f,g)$ =NO otherwise. In the worst case setting, we wish to find an approximation operator U such that

$$U(f,g) = \mathrm{VER}(f,g) \qquad \forall f \in F,\ g \in G.$$

In the probabilistic setting, we assume that the set F is equipped with a probability measure μ. For a given confidence parameter $\delta \in [0,1]$, we wish to find an approximation operator U such that

$$\mu\{f \in F;\ U(f,g) = \mathrm{VER}(f,g)\} \ge 1 - \delta, \qquad \forall g \in G.$$

For relaxed verification, we assume that $\alpha \in [0,1]$ and we redefine $\mathrm{VER}(f,g)$ as follows. We set $\mathrm{VER}(f,g)$ =YES if $\|S(f) - g\| \le \varepsilon$, $\mathrm{VER}(f,g)$ =NO if $\|S(f - g\| > (1+\alpha)\varepsilon$, and $\mathrm{VER}(f,g)$ =DON'T CARE, otherwise.

The complexity of verification or relaxed verification is defined similarly as for computational problems, that is, by minimizing the cost of computing U that solves the corresponding verification problem.

Acknowledgments

We thank Z. Galil, G. W. Wasilkowski and A. G. Werschulz for valuable comments.

REFERENCES

Bakhvalov, N. S. [59], *On approximate calculation of integrals (in Russian)*, Vestnik MGV, Ser. Mat. Mekh. Astron. Fiz. Khim. 4 (1959), 3–18.

Bakhvalov, N. S. [71], *On the optimality of linear methods for operator approximation in convex classes of functions (in Russian)*, Zh. Vychisl. Mat. Mat. Fiz. **11** (1971), 1014–1018 English transl. USSR Comput. Math. Math. Phys., 11, 1971, 244–249.

Beck, J. and Chen, W. W. L. [87], *Irregularities of Distribution*. Cambridge University Press, 1987.

Blum, L., Shub, M. and Smale, S. [89], *On a Theory of Computation and Complexity over the Real Numbers: NP-Complete- ness, Recursive Functions and Universal Machines*, Bull. Amer. Math. Soc. **21** (1989), 1–46.

Chu, M. [94], *There exists a problem whose computational complexity is any given function of the information complexity*, to appear in J. Complexity (1994).

Dobkin, D. P. and Mitchell, D. P. [93], *Random-Edge Discrepancy of Supersampling Patterns*, Graphics Interface '93, York, Ontario (1993).

Golomb, M. and Weinberger, H. F. [59], *Optimal approximation and error bounds*, in, *On Numerical Approximation (R. E. Langer, ed.)*, Univ. of Wisconsin Press, Madison, 1959, pp. 117–190.

Heinrich, S. [92], *Random Approximation in Numerical Analysis*, Proc. of the Functional Analysis Conference, Essen 1991, "Lecture Notes in Pure and Applied Mathematics", Marcel Dekker (1992).

Heinrich, S. [93], *Complexity of integral equations and relations to s-numbers*, J. Complexity **9** (1993), 141–153.

Kacewicz, B. Z. and Plaskota, L. [90], *On the minimal cost of approximating linear problems based on information with deterministic noise*, Numer. Funct. Anal. and Optimiz. **11** (1990), 511–528.

Kiefer, J. [53], *Sequential minimax search for a maximum*, Proc. Amer. Math. Soc. **4** (1953), 502–505.

Ker-I Ko [91], *Complexity Theory of Real Functions*, Birkhäser, Boston, Mass., 1991.

Micchelli, C. A. and Rivlin, T. J. [77], *A survey of optimal recovery*, in Optimal Estimation in Approximation Theory (C. A. Micchelli and T. J. Rivlin, eds.), Plenum, New York (1977).

Nemirovsky, A. S. [91], *On optimality of Krylov's information when solving linear operator equations*, J. Complexity **7** (1991), 121–130.

Nemirovsky, A. S. [92], *Information-Based Complexity of Linear Operator Equations*, J. Complexity **8** (1992), 153–175.

Nemirovsky, A. S. and Yudin, D. B. [83], *Problem Complexity and Method Efficiency in Optimization*, Wiley-Interscience, New York, 1983.

Niederreiter, H. [92], *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, Philadelphia, Penn., 1992.

Nikolskij, S. M. [50], *On the problem of approximation estimate by quadrature formulas (in Russian),*, Usp. Mat. Nauk **5** (1950), 165–177.

Novak, E. [88], *Deterministic and Stochastic Error Bounds in Numerical Analysis*, Lectures Notes in Math. Springer Verlag, Berlin, **1349**, 1988.

Novak, E. [92], *Optimal linear randomization methods for linear operators in Hilbert spaces*, J. Complexity **8** (1992), 22-36.

Novak, E. [93], *Algorithms and complexity for continuous problems*, to appear in "Geometry, Analysis, and Mechanics" J. M. Rassias, ed., World Scientific, Singapore (1993).

Novak, E. and Woźniakowski, H. [92], *Relaxed Verification for Continuous Problems*, J. Complexity **8** (1992), 124-152.

Packel, E. W. and Traub, J. F. [87], *Information-based complexity*, Nature **328** (1987), 29-33.

Packel, E. W. and Woźniakowski, H. [87], *Recent developments in information-based complexity*, Bull. Amer. Math. Soc. **17** (1987), 9-36.

Papadimitriou, C. H. and Tsitsiklis, J. [86], *Intractable problems in control theory*, SIAM J. Control Optim. **24** (1986), 639-654.

Paskov, S. H. [93], *Average Case Complexity of Multivariate Integration for Smooth Functions*, J. Complexity **9** (1993), 291-312.

Pereverzev, S. V. [89], *On the complexity of the problem of finding solutions of Fredholm equations of the second kind with differentiable kernels (in Russian)*, Ukrain. Mat. Sh. **41** (1989), 1422-1425.

Roth, K. F. [54], *On irregularities of distribution*, Mathematika **1** (1954), 73-79.

Roth, K. F. [80], *On irregularities of distribution, IV*, Acta Arith. **37** (1980), 67-75.

Sard, A. [49], *Best approximate integration formulas; best approximation formulas*, Amer. J. Math. **71** (1949), 80-91.

Schönhage, A. [86], *Equation solving in terms of computational complexity*, Proc. Intern. Congress Math., Berkeley (1986).

Schoenberg, I. J. [64], *Spline interpolation and best quadrature formulas*, Bull. Amer. Math. Soc. **70** (1964), 143-148.

Traub, J, F. [61], *On functional iteration and the calculation of roots*, Preprints of papers, 16th Natl. ACM Conf. Session 5A-1, Los Angeles, Ca. (1961), 1-4.

Traub, J, F. [64], *Iterative Methods for the Solution of Equations*, Englewood Cliffs, N. J., 1964. Reissued: Chelsea Press, New York, 1982.

Traub, J. F., Wasilkowski, G. W. and Woźniakowski, H. [83], *Information, Uncertainty, Complexity*, Addison-Wesley, Reading, Mass., 1983.

Traub, J. F., Wasilkowski, G. W. and Woźniakowski, H. [88], *Information-Based Complexity*, Academic Press, New York, 1988.

Traub, J. F. and Woźniakowski, H. [80], *A General Theory of Optimal Algorithms*, Academic Press, New York, 1980.

Traub, J. F. and Woźniakowski, H. [82], *Complexity of linear programming*, Oper. Res. Lett. **1** (1982), 59-62.

Traub, J. F. and Woźniakowski, H. [84], *On the optimal solution of large linear systems*, J. Assoc. Comput. Mach. **31** (1984), 545-559.

Traub, J. F. and Woźniakowski, H. [91a], *Theory and Applications of Information-Based Complexity*, 1990 Lectures in Complex Systems, Santa Fe Institute,, 163-193, Addison-Wesley, Reading, Mass., 1991.

Traub, J. F. and Woźniakowski, H. [91b], *Information-Based Complexity: New Questions for Mathematicians*, Math. Intell. **13** (1991), 34-43.

Wasilkowski, G. W. [89a], *A Clock Synchronization Problem with Random Delays*, J. Complexity **5** (1989), 1-11.

Wasilkowski, G. W. [89b], *Randomization for Continuous Problems*, J. Complexity **5** (1989), 195–218.

Wasilkowski, G. W. [93], *Integration and approximation of multivariate functions: average case complexity with isotropic Wiener measure*, Bull. Amer. Math. Soc. (N.S) **28** (1993), 308–314.

Wasilkowski, G.W. and Woźniakowski, H. [93], *There exists a linear problem with infinite combinatory complexity*, J. Complexity **9** (1993), 326–337.

Wasilkowski, G.W. and Woźniakowski, H. [94], in progress (1994).

Werschulz, A. G. [91], *The Computational Complexity of Differential and Integral Equations*, Oxford University Press, Oxford, 1991.

Woźniakowski, H. [75], *Generalized information and maximal order of iteration for operator equations*, SIAM J. Numer. Anal. **12** (1975), 121–135.

Woźniakowski, H. [91], *Average Case Complexity of Multivariate Integration* , Bull. Amer. Math. Soc. (N.S) **24** (1991), 185–194.

Woźniakowski, H. [92a], *Complexity of Verification and Computation for IBC Problems*, J. Complexity **8** (1992), 93–123.

Woźniakowski, H. [92b], *Average case complexity of linear multivariate problems, Part 1: Theory, Part 2: Applications*, J. Complexity **8** (1992), 337–372, 373–392.

Woźniakowski, H. [93], *Tractability and Strong Tractability of Linear Multivariate Problems*, in progress (1993).

SCIENTIFIC
AMERICAN

Breaking Intractability

by Joseph F. Traub and Henryk Woźniakowski

SCIENTIFIC
AMERICAN
JANUARY 1994
VOL. 270, NO. 1 PP. 102-107

Breaking Intractability

Problems that would otherwise be impossible to solve can now be computed, as long as one settles for what happens on the average

by Joseph F. Traub and Henryk Woźniakowski

Although mathematicians and scientists must rank among the most rational people in the world, they will often admit to falling prey to a curse. Called the curse of dimension, it is one many people experience in some form. For example, a family's decision about whether to refinance their mortgage with a 15- or 30-year loan can be extremely difficult to make, because the choice depends on an interplay of monthly expenses, income, future tax and interest rates and other uncertainties. In science, the problems are more esoteric and arguably much harder to cope with. In the computer-aided design of pharmaceuticals, for instance, one might need to know how tightly a drug candidate will bind to a biological receptor. Assuming a typical number of 8,000 atoms in the drug, the biological receptor and the solvent, then because of the three spatial variables needed to describe the position of each atom, the calculation involves 24,000 variables. Simply put, the more variables, or dimensions, there are to consider, the harder it is to accomplish a task. For many problems, the difficulty grows exponentially with the number of variables.

A potentially intractable problem

The curse of dimension can elevate tasks to a level of difficulty at which they become intractable. Even though scientists have computers at their disposal, problems can have so many variables that no future increase in computer speed will make it possible to solve them in a reasonable amount of time.

Can intractable problems be made tractable—that is, solvable in a relatively modest amount of computer time? Sometimes the answer is, happily, yes. But we must be willing to do without a guarantee of achieving a small error in our calculations. By settling for a small error most of the time (rather than always), some kinds of multivariate problems become tractable. One of us (Woźniakowski) formally proved that such an approach works for at least two classes of mathematical problems that arise quite frequently in scientific and engineering tasks. The first is integration, a fundamental component of the calculus. The second is surface reconstruction, in which pieces of information are used to reconstruct an object, a technique that is the basis for medical imaging.

Fields other than science can benefit from ways of breaking intractability. For example, financial institutions often have to assign a value to a pool of mortgages, which is affected by mortgagees who refinance their loans. If we assume a pool of 30-year mortgages and permit refinancing monthly, then this task contains 30 years times 12 months, or 360 variables. Adding to the difficulty is that the value of the pool depends on interest rates over the next 30 years, which are of course unknown.

We shall describe the causes of intractability and discuss the techniques that sometimes allow us to break it. This issue belongs to the new field of information-based complexity, which examines the computational complexity of problems that cannot be solved exactly. We shall also speculate briefly on how information-based complexity might enable us to prove that certain scientific questions can never be answered because the necessary computing resources do not exist in the universe. If so, this condition would set limits on what is scientifically knowable.

Information-based complexity focuses on the computational difficulty of so-called continuous problems. Calculating the movement of the planets is an example. The motion is governed by a system of ordinary differential equations—that is, equations that describe the positions of the planets as a function of time. Because time can take any real value, the mathemati-

JOSEPH F. TRAUB and HENRYK WOŹNIAKOWSKI have been collaborating since 1973. Currently the Edwin Howard Armstrong Professor of Computer Science at Columbia University, Traub headed the computer science department at Carnegie Mellon University and was founding chair of the Computer Science and Telecommunications Board of the National Academy of Sciences. In 1959 he began his pioneering research in what is today called information-based complexity and has received many honors, including election to the National Academy of Engineering. He is grateful to researchers at the Santa Fe Institute for numerous stimulating conversations concerning the limits of scientific knowledge. Woźniakowski holds two tenured appointments, one at the University of Warsaw and the other at Columbia University. He directed the department of mathematics, computer science and mechanics at the University of Warsaw and was the chairman of Solidarity there. In 1988 he received the Mazur Prize from the Polish Mathematical Society. The authors thank the National Science Foundation and the Air Force Office of Scientific Research for their support.

cal model is said to be continuous. Continuous problems are distinct from discrete problems, such as difference equations in which time takes only integer values. Difference equations appear in such analyses as the predicted number of predators in a study of predator-prey populations or the anticipated pollution levels in a lake.

In the everyday process of doing science and engineering, however, continuous mathematical formulations predominate. They include a host of problems, such as ordinary and partial differential equations, integral equations, linear and nonlinear optimization, integration and surface reconstruction. These formulations often involve a large number of variables. For example, computations in chemistry, pharmaceutical design and metallurgy often entail calculations of the spatial positions and momenta of thousands of particles.

Often the intrinsic difficulty of guaranteeing an accurate numerical solution grows exponentially with the number of variables, eventually making the problem computationally intractable. The growth is so explosive that in many cases an adequate numerical solution cannot be guaranteed for situations comprising even a modest number of variables.

To state the issue of intractability more precisely and to discuss possible cures, we will consider the example of computing the area under a curve. The process resembles the task of computing the vertical area occupied by a collection of books on a shelf. More explicitly, we will calculate the area between two bookends. Without loss of generality, we can assume the bookends rest at 0 and 1. Mathematically, this summing process is called the computation of the definite integral. (More accurately, the area is occupied by an infinite number of books, each infinitesimally thin.) The mathematical input to this problem is called the integrand, a function that describes the profile of the books on the shelf.

Calculus students learn to compute the definite integral by following a set of prescribed rules. As a result, the students arrive at the exact answer. But most integration problems that arise in practice are far more complicated, and the symbolic process learned in school cannot be carried out. Instead the integral must be approximated numerically—that is, by a computer. More exactly, one computes the integrand values at finitely many points. These integrand values result from so-called information operations. Then one combines these values to produce the answer.

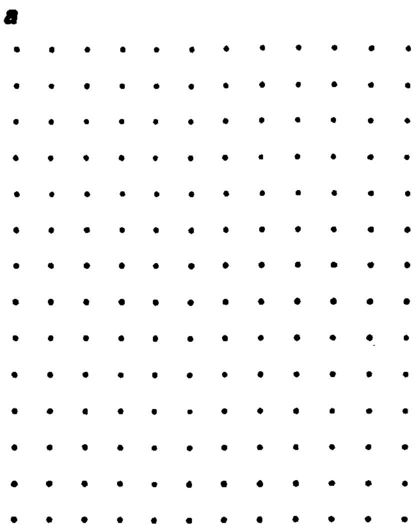
Knowing only these values does not completely identify the true integrand. Because one can evaluate the integrand only at a finite number of points, the information about the integrand is partial. Therefore, the integral can, at best, only be approximated. One typically specifies the accuracy of the approximation by stating that the error of the answer falls within some error threshold. Mathematicians represent this error with the Greek letter epsilon, ε.

Even this goal cannot be achieved without further restriction. Knowing the integrand at, say, 0.2 and 0.5 indicates nothing about the curve between those two points. The curve can assume any shape between them and therefore enclose any area. In our bookshelf analogy, it is as if an art book has been shoved between a run of paperbacks. To guarantee an error of at most ε, some global knowledge of the integrand is needed. One may need to assume, for example, that the slope of the function is always less than 45 degrees—or that only paperbacks are allowed on that shelf.

In summary, an investigator trying to solve an integral must usually do it numerically on a computer. The input to the computer is the integrand values at some points. The computer produces an output that is a number approximating the integral.

The basic concept of computational complexity can now be introduced. We want to find the intrinsic difficulty of solving the integration problem. Assume that determining integrand values and using combinatory operations, such as addition, multiplication and comparison, each have a given cost. The cost could simply be the amount of time a computer needs to perform the operation. Then the computational complexity of this integra-

One solution to an intractable problem



SAMPLING POINTS indicate where to evaluate functions in the randomized and average-case settings. The points are plotted in two dimensions for visual clarity. The points chosen can be spaced over regular intervals such as grid points (*a*), or in random positions (*b*). Two other types, so-called Hammersley points (*c*) and hyperbolic-cross points (*d*), represent optimal places in the average-case setting.

tion problem can be defined as the minimal cost of guaranteeing that the computed answer is within an error threshold, ε, of the true value. The optimal information operations and the optimal combinatory algorithm are those that minimize the cost.

Theorems have shown that the computational complexity of this integration problem is on the order of the reciprocal of the error threshold ($1/\varepsilon$). In other words, it is possible to choose a set of information operations and a combinatory algorithm such that the solution can be approximated at a cost of about $1/\varepsilon$. It is impossible to do better. With one variable, or dimension, the problem is rather easy. The computational complexity is inversely proportional to the desired accuracy.

But if there are more dimensions to this integration problem, then the computational complexity scales exponentially with the number of variables. If *d* represents the number of variables, then the complexity is on the order of $(1/\varepsilon)^d$—that is, the reciprocal of the error threshold raised to a power equal to the number of variables. If one wants eight-place accuracy (down to 0.00000001) in computing an integral that has three variables, then the complexity is roughly 10^{24}. In other words, it would take a trillion trillion integrand values to achieve that level of accuracy. Even if one generously assumes the existence of a sequential computer that performs 10 billion function evaluations per second, the job would take 100 trillion seconds, or more than three million years. A computer with a million processors would still take 100 million seconds, or about three years.

To discuss multivariate problems more generally, we must introduce one additional parameter, called *r*. This parameter represents the smoothness of the mathematical inputs. By smoothness, we mean that the inputs consist of functions that do not have any sudden or dramatic changes. (Mathematicians say that all partial derivatives of the function up to order *r* are bounded.) The parameter takes on nonnegative integer values; increasing values indicate more smoothness. Hence, $r = 0$ represents the least amount of smoothness (technically, the integrands are only continuous—they are rather jagged but still connected as a single curve).

Numerous problems have a computational complexity that is on the order of $(1/\varepsilon)^{d/r}$. For those of a more technical persuasion, multivariate integration, surface reconstruction, partial differential equations, integral equations and nonlinear optimization all have this computational complexity.

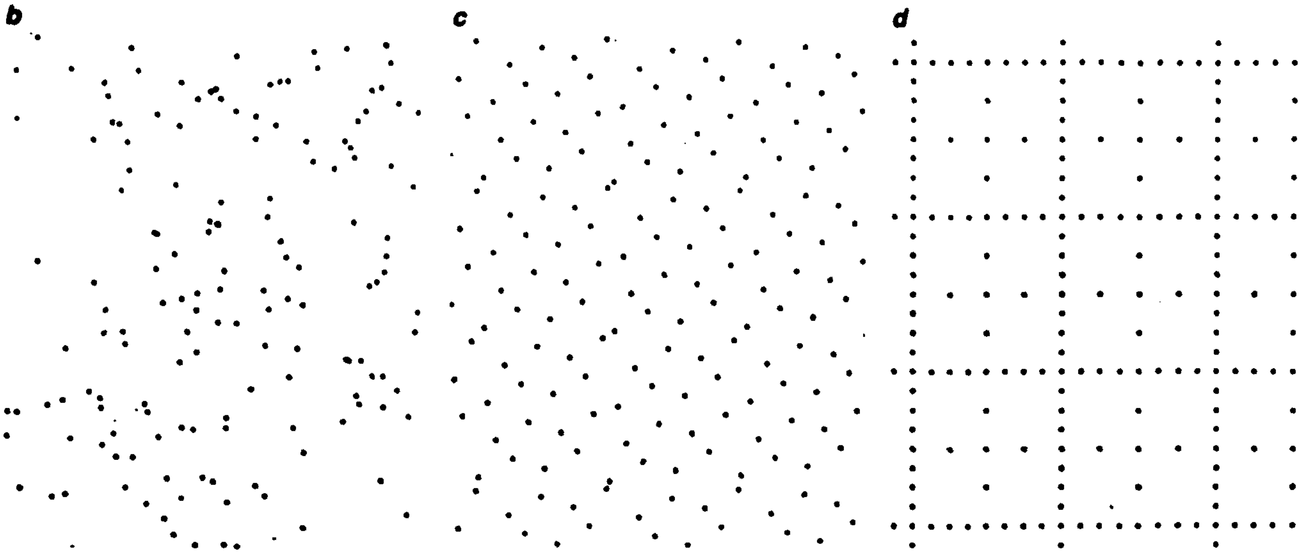
If the error threshold and the smoothness parameter are fixed, then the computational complexity depends exponentially on the number of dimensions. Hence, the problems become intractable for high dimensions. An impediment even more serious than intractability may occur: a problem may be unsolvable. A problem is unsolvable if one cannot compute even an approximation at finite cost. This is the case when the mathematical inputs are continuous but jagged. The smoothness pa-

Developing a Random Approach

In the 1940s physicists working on the Manhattan Project at Los Alamos National Laboratory realized that some of the problems they were trying to solve, such as the movement of neutrons through materials, lay beyond the reach of deterministic calculations. They turned to the Monte Carlo method of Nicholas C. Metropolis and Stanislaw M. Ulam. The strength of the method is that its error does not depend on the number of variables in the problem. Hence, if applicable, it breaks the curse of dimension. The classical Monte Carlo method for multivariate integration requires at most of order $1/\varepsilon^2$ evaluations at random points, where ε is the error bound. An alternative statement is that if the integrand is evaluated at *n* random points, then the expected error of randomization is at most of order $1/\sqrt{n}$. Since its formulation, the Monte Carlo method and its variations have proved to be useful to calculate a variety of phenomena, from the size of cosmic showers to the percolation of a liquid through a solid.

Stanislaw M. Ulam, 1909-84

For multivariate integration, the classical Monte Carlo method is optimal only if the smoothness parameter, *r*, of integrands is zero. In 1959 the Russian mathematician N. S. Bakhvalov began pioneering research on the computational complexity of multivariate integration in the randomized setting and devised an alternative to the Monte Carlo method. Later, in 1988, Erich Novak of the University of Erlangen-Nürnberg extended the work of Bakhvalov to establish that the computational complexity in the randomized setting is of order $(1/\varepsilon)^s$, with $s = 2/(1 + 2r/d)$. Note that $0 < s \leq 2$. If the smoothness parameter equals zero, then $s = 2$, and the classical Monte Carlo method is optimal. On the other hand, if *r* is positive, then the classical Monte Carlo method is no longer optimal, and Bakhvalov's method can be used instead.



rameter is zero, and the computational complexity becomes infinite. Hence, for many problems with a large number of variables, guaranteeing that an approximation has a desired error becomes an unsolvable or intractable task.

Mathematically, the computational complexity results we have described apply to the so-called worst-case deterministic setting. The "worst case" phrasing comes from the fact that the approximation provides a guarantee that the error always falls within ε. In other words, for multivariate integration, an approximation within the error threshold is guaranteed for every integrand that has a given smoothness. The word "deterministic" arises from the fact that the integrand is evaluated at deterministic (in contrast to random) points.

In this worst-case deterministic setting, many multivariate problems are unsolvable or intractable. Because these results are intrinsic to the problem, one cannot get around them by inventing other methods.

One possible way to break unsolvability and intractability is through randomization. To illustrate how randomization works, we will again use multivariate integration. Instead of picking points deterministically or even optimally, we allow (in an informal sense) a coin toss to make the decisions for us. A loose analogy might be sampling polls. Rather than ask every registered voter, a pollster conducts a small, random sampling to determine the likely winner.

Theorems indicate that with a random selection of points, the computational complexity is at most on the order of the reciprocal of the square of the error threshold ($1/\varepsilon^2$). Thus, the problem is always tractable, even if the smoothness parameter is equal to zero.

The workhorse of the randomized approach has been the Monte Carlo method. Nicholas C. Metropolis and Stanislaw M. Ulam suggested the idea in the 1940s. In the classical Monte Carlo method the integrand is evaluated at uniformly distributed random points. The arithmetic mean of these function values then serves as the approximation of the integral.

Amazingly enough, for multivariate integration problems, randomization of this kind makes the computational complexity independent of dimension. Problems that are unsolvable or intractable if computed from the best possible deterministic points become tractable if approached randomly. (If r is positive, however, then the classical Monte Carlo method is not the optimal one; see box on the opposite page.)

One does not get so much for nothing. The price that must be paid for breaking the unsolvability or intractability is that the ironclad guarantee that the error is at most ε is lost. Instead one is left only with a weaker guarantee that the error is probably no more than ε—much as a preelection poll is usually correct but might, on occasion, predict a wrong winner. In other words, a worst-case guarantee is impossible; one must be content with a weaker assurance.

Randomization makes multivariate integration and many other important problems computationally feasible. It is not, however, a cure-all. Randomization fails completely for some kinds of problems. For instance, in 1987 Greg W. Wasilkowski of the University of Kentucky showed that randomization does not break intractability for surface re-

Average-Case Complexity

In the text, we mention that the average-case complexity of multivariate integration is on the order of the reciprocal of the error threshold ($1/\varepsilon$) and that for surface reconstruction, it is the square of that reciprocal ($1/\varepsilon^2$). For simplicity, we ignored some multiplicative factors that depend on d and ε. Here we provide more rigorous statements.

The average computational complexity, $\mathrm{comp}^{\mathrm{avg}}(\varepsilon, d; \mathrm{INT})$, of multivariate integration is bounded by

$$\frac{g_1(d)}{\varepsilon}\left(\log\frac{1}{\varepsilon}\right)^{(d-1)/2} \leq \mathrm{comp}^{\mathrm{avg}}(\varepsilon, d; \mathrm{INT}) \leq \frac{g_2(d)}{\varepsilon}\left(\log\frac{1}{\varepsilon}\right)^{(d-1)/2}$$

The average computational complexity, $\mathrm{comp}^{\mathrm{avg}}(\varepsilon, d; \mathrm{SUR})$, of surface reconstruction is bounded by

$$\frac{g_3(d)}{\varepsilon^2}\left(\log\frac{1}{\varepsilon}\right)^{2(d-1)} \leq \mathrm{comp}^{\mathrm{avg}}(\varepsilon, d; \mathrm{SUR}) \leq \frac{g_4(d)}{\varepsilon^2}\left(\log\frac{1}{\varepsilon}\right)^{2(d-1)}$$

Good estimates of $g_1(d)$, $g_2(d)$, $g_3(d)$ and $g_4(d)$ are currently not known.

construction. Is there an approach that does and that works over a broad class of mathematics problems?

There is indeed. It is the average-case setting, in which we seek to break unsolvability and intractability by replacing a worst-case guarantee with a weaker one: that the expected error is at most ε. The average-case setting imposes restrictions on the kind of mathematical inputs. These restrictions are chosen to represent what would happen most of the time. Technically, the constraints are described by probability distributions; the distributions describe the likelihood that certain inputs occur. The most commonly used distributions are Gaussian measures and, in particular, Wiener measures.

Although it was known since the 1960s that multivariate integration is tractable on the average, the proof was nonconstructive. That is, it did not specify the optimal points to evaluate the integrand, the optimal combinatory algorithm and the average computational complexity. Attempts to apply ideas from other areas of computation to determine these unknowns did not work. For example, evaluating the integrand at regularly spaced points, such as those on a grid, are often used in computation. But theorems have shown them to be poor choices for the average-case setting. A proof was given in 1975 by N. Donald Ylvisaker of the University of California at Los Angeles. It was later generalized in 1990 by Wasilkowski and Anargyros Papageorgiou, then studying for his Ph.D. at Columbia University.

The solution came in 1991, when Woźniakowski found the construction. As sometimes happens in science, a result from number theory, a branch of mathematics far removed from average-case complexity theory, was crucial. Part of the key came from work on number theory by Klaus F. Roth of Imperial College, London, a 1958 Fields Medalist. Another part was provided by recent work by Wasilkowski.

Let us describe the result more precisely. First, put the smoothness parameter at zero—that is, tackle a problem that is unsolvable in the worst-case deterministic setting. Next, assume that integrands are distributed according to a Wiener measure. If we ignore certain multiplicative factors for simplicity's sake, the average computational complexity has been proved to be inversely proportional to the error threshold (on the order of $1/\varepsilon$) [*see box on page 5*]. For small errors, the result is a major improvement over the classical Monte Carlo method, in which the cost is inversely proportional to the square of the error threshold ($1/\varepsilon^2$).

The average case offers a different kind of assurance from that provided by the randomized (Monte Carlo) setting. The error in the average-case setting depends on the distribution of the integrands, whereas the error in the randomized setting depends on a distribution of the sample points. In our books-on-a-shelf analogy, the distribution in the average-case setting might rule out the inclusion of many oversize books, whereas the distribution in the randomized setting determines which books are to be sampled.

In the average-case setting the optimal evaluation points must be deterministically chosen. The best points are Hammersley points or hyperbolic-cross points [*see illustration on pages 4 and 5*]. These deterministic points offer a better sampling than randomly selected or regularly spaced (or grid) points. They make what would be impossible to solve tractable on average.

Is surface reconstruction also tractable on the average? This query is particularly important because, as already mentioned, randomization does not help. Under the same assumptions we used for integration, we find that the average computational complexity is on the order of $1/\varepsilon^2$. Hence, surface reconstruction becomes tractable on average. As was the case for integration, hyperbolic-cross points are optimal.

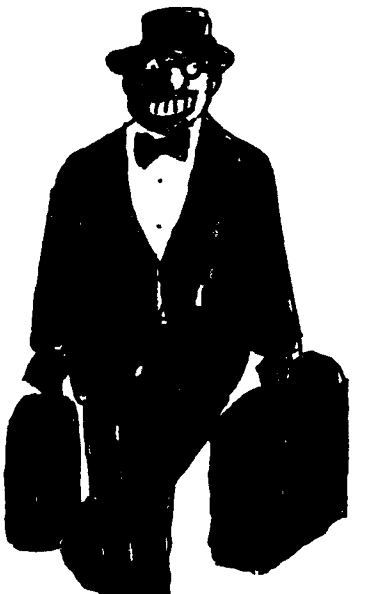
We are now testing whether the average case is a practical alternative. A Ph.D. student at Columbia, Spassimir H. Paskov, is developing software to compare the deterministic techniques with Monte Carlo methods for integration. Preliminary results obtained by testing certain finance problems suggest the superiority of the deterministic methods in practice.

In our simplified description, we ignored a multiplicative factor that affects the computational complexity. This factor depends on the number of variables in the problem. When the number of variables is large, that factor can become huge. Good theoretical estimates of the factor are not known, and obtaining them is believed to be very hard.

Woźniakowski uncovered a solution: get rid of that factor. Specifically, we say a problem is strongly tractable if the number of function evaluations needed

Discrete Computational Complexity

This article discusses intractability and breaking of intractability for multivariate integration and surface reconstruction. These are two examples of continuous problems. But what is known about the computational complexity of discrete, rather than continuous, problems? The famous traveling salesman problem is an example of a discrete problem, in which the goal is to visit various cities in the shortest distance possible.



A discrete problem is intractable if its computational complexity increases exponentially with the number of its inputs. The intractability of many discrete problems in the worst-case deterministic setting has been conjectured but not yet proved. What is known is that hundreds of discrete problems all have essentially the same computational complexity. That means they are all tractable or all intractable, and the common belief among experts is that they are all intractable. For technical reasons, these problems are said to be NP-complete. One of the great open questions in discrete computational complexity theory is whether the NP-complete problems are indeed intractable [see "Turing Machines," by John E. Hopcroft; SCIENTIFIC AMERICAN, May 1984].

for the solution is completely independent of the number of variables. Instead it would depend only on a power of 1/ε. The possibility seems too much to hope for, but it was proved last year that multivariate integration and surface reconstruction are both strongly tractable on the average.

A final obstacle must be overcome before these new results can be used. We know there must exist evaluation points and a combinatory algorithm that make integration and surface reconstruction strongly tractable on the average. Unfortunately, the proof of this result does not tell us what the points and algorithms are, thus leaving a beautiful challenge for the future.

REENTRY OF SPACE SHUTTLE provides an example of a computationally complex task: modeling of the airflow around the craft. This job is difficult even though only seven variables govern the dynamics. Added dimensions may yield problems that can never be solved and thus limit what is scientifically knowable.

Work on information-based complexity has led one of us (Traub) to speculate that it might be possible to prove formally that certain scientific questions are unanswerable. The proposed attack is to prove that the computing resources (time, memory, energy) do not exist in the universe to answer such questions.

One important achievement of mathematics over the past 60 years is the idea that mathematical problems may be undecidable, noncomputable or intractable. Kurt Gödel proved the first of these results. He established that in a sufficiently rich mathematical system, such as arithmetic, there are theorems that can never be proved.

We believe it is time to up the ante and try to prove there are unanswerable scientific questions. In other words, we would like to establish a physical Gödel's theorem. The process offers a markedly different challenge from proving results about mathematical problems, because a scientific question does not come equipped with a mathematical formulation. Such questions include when the universe will stop expanding and what the average global temperature will be in the year 2001.

Why do intractability results suggest that some scientific questions might be unanswerable? Recall the results. In the worst-case deterministic setting, the computational complexity of many continuous problems grows exponentially with dimension. Also, the computational complexity of many discrete problems is conjectured to grow exponentially with the number of inputs [*see box on opposite page*]. Furthermore, although some problems are tractable in the randomized or average-case settings, it has been proved that others remain intractable. Such problems may lurk in certain supercomputing tasks or questions regarding the foundations of physics. After all, they involve a large number of variables or particles. Even worse, many physics problems require solutions to a kind of integral called a path integral, which has an infinite number of dimensions. Solutions of path integrals invite high-dimensional approximations. Thus, the intractability results and conjectures are certainly daunting because they suggest that many tasks with a large number of variables or objects might be impossible to solve.

We emphasize the possibility of other impediments to answering scientific questions. One is chaos, the extreme sensitivity to initial conditions. Because the precise initial conditions are either not known or cannot be exactly entered into a digital computer, certain questions about the behavior of a chaotic system cannot be answered. To focus on the issue at hand, we limit ourselves to intractability.

As we have already observed, a scientific question does not come equipped with a mathematical formulation. Each of a number of models might capture the essence of a scientific question. Because intractability results refer to a particular mathematical formulation, it might happen that although a particular mathematical formulation is intractable, another formulation may be found that is indeed tractable. This prospect indicates a possible way to prove the existence of unanswerable scientific questions. We can attempt to show that there exist scientific questions such that every mathematical formulation that captures the essence of the question is intractable. We would therefore have science's version of Gödel's theorem.

Humans are intrigued not only by the unknown but also by the unknowable. Here we have suggested one possible attack to establish what may be forever unknowable in science. The curse of dimension, broken now for many kinds of problems, may yet cast its spell.

FURTHER READING

INFORMATION-BASED COMPLEXITY. E. W. Packel and J. F. Traub in *Nature*, Vol. 328, No. 6125, pages 29–33; July 2, 1987.

INFORMATION-BASED COMPLEXITY. J. F. Traub, G. W. Wasilkowski and H. Woźniakowski. Academic Press, 1988.

AVERAGE CASE COMPLEXITY OF MULTIVARIATE INTEGRATION. H. Woźniakowski in *Bulletin of the American Mathematical Society*, Vol. 24, No. 1, pages 185–194; January 1991.

THE COMPUTATIONAL COMPLEXITY OF DIFFERENTIAL AND INTEGRAL EQUATIONS: AN INFORMATION-BASED APPROACH. Arthur G. Werschulz. Oxford University Press, 1991.

THEORY AND APPLICATIONS OF INFORMATION-BASED COMPLEXITY. J. F. Traub and H. Woźniakowski in *1990 Lectures in Complex Systems, Santa Fe Institute.* Edited by Lynn Nadel and Daniel L. Stein. Addison-Wesley, 1991.

WHAT IS SCIENTIFICALLY KNOWABLE? J. F. Traub in *Carnegie Mellon University Computer Science: A 25th Anniversary Commemorative.* Edited by Richard F. Rashid. Addison-Wesley, 1991.